

Claims

Sub A1

1. A system for developing software, comprising:
 - a version store for storing a plurality of development documents;
 - a plurality of service providers performing different development functions, and each having a mutually compatible interface;
 - a client having the same compatible interface, for receiving commands from a user and for routing them to various ones of the service providers for execution upon the development documents from the version store.
2. The system of claim 1 where all of the service providers and the client are replaceable by a third party.
3. The system of claim 1 where the version store is a database.
4. The system of claim 1 where the interface comprises an API having a set of methods.
5. The system of claim 4 where at least one of the service providers implements less than all of the methods in the set.
6. The system of claim 4 where the API includes at least a subset of OLE DB.
7. The system of claim 1 further including an object model also exposing the same compatible interface.
8. The system of claim 1 further including a command-line utility for passing user commands to service providers through the same compatible interface.
9. The system of claim 1 where the version store and the client are physically located in different computers.

*Sub
an*

10. The system of claim 1 further comprising a merge subsystem for providing merge/differencing services.
11. The system of claim 1 further comprising a keyword subsystem for providing keyword-expansion services.
12. A service provider for a software development system, comprising:
code for performing a development service in a software development system;
an interface for communicating with a set of further service providers and with a client for receiving development commands from a user.
13. The service provider of claim 12 where the provider also communicates with a version store for storing and retrieving development documents.
14. The service provider of claim 13 where the commands involve processing one or more of the development documents from the version store.
15. The service provider of claim 12 where the development service is query processing.
16. The service provider of claim 12 where the development service is enlistment management.
17. The service provider of claim 16 further including
a private store for holding copies of some of the development documents.
an interface to the private store for storing and retrieving the copies.
18. The service provider of claim 16 where the interface to the private store is compatible with the interface for communicating to the further service providers.
19. The service provider of claim 16 where the private store is physically a part of the version store.

✓ all

20. The service provider of claim 12 where the development service is query processing.
21. The service provider of claim 12 where the interface comprises an API having a set of methods.
22. The service provider of claim 21 where at least one of the other service providers implements less than all of the methods in the set.
23. The service provider of claim 21 where the set of methods is a subset of a set of methods implemented by at least one of the other service providers.
24. The service provider of claim 21 where the API includes at least a subset of OLE DB.
25. A client for a software development system having a plurality of service providers, comprising:
 - means for receiving commands from a user for executing development operations;
 - an interface for communicating with compatible interfaces of all the plurality of service providers.
26. The client of claim 25 further including means for accessing and processing a plurality of enumerators representing service providers available to the system.
27. The client of claim 25 further comprising an interface to a merge subsystem.
28. The client of claim 27 where the merge subsystem comprises:
 - a plurality of preprocessors for receiving and modifying an input stream representing a plurality of input development documents;
 - a merge engine for producing an output merge document from the input documents.

Draft

29. The client of claim 28 where the merge subsystem further comprises a merge broker for selecting one of the preprocessors.

30. The client of claim 28 where the merge subsystem further includes a plurality of postprocessors for processing the output merge document.

31. The client of claim 25 further comprising an interface to a keyword processing subsystem.

32. The client of claim 31 where the keyword subsystem comprises:
a plurality of expanders for processing different sets of keywords;
a keyword broker for selecting among the expanders.

33. The client of claim 25 where at least some of the service providers also communicate with a version store for storing and retrieving development documents.

34. A merge subsystem for a software development system, comprising:
a merge engine for merging a plurality of development documents into a merge document;
a plurality of merge preprocessors for modifying the merge documents and passing them to the merge engine;
a merge broker for receiving a merge command and for selecting one of the preprocessors in response to the command.

35. The merge subsystem of claim 34 where the command specifies the development documents.

36. The merge subsystem of claim 35 where the command includes a parameter for selecting the one preprocessor.

*Dick
all*

37. The merge subsystem of claim 34 further comprising a plurality of postprocessors for modifying the merge document.

38. The merge subsystem of claim 37 further including means for selecting one of the postprocessors.

39. A keyword expansion subsystem for a software development system, comprising:
a plurality of keyword expanders for expanding keywords in a development document;
a keyword broker for receiving a command to expand the development document and for selecting one of the providers in response to the command.

40. The keyword expansion subsystem of claim 39 where a parameter in the command specifies a keyword domain for the development document.

41. A method for developing software using versioned documents in a programmed digital computer, comprising:
sending commands from a client through a versioning interface;
receiving the commands in a plurality of service providers all having an interface compatible with the versioning interface;
executing the commands in the service providers;
in the course of executing the commands, retrieving and storing a plurality of development documents in a version store.

42. The method of claim 41 where the development documents are retrieved to and stored from at least some of the service providers.

43. The method of claim 42 where the service providers communicate with the version store via an interface different from the versioning interface.

44. The method of claim 41 where all of the interfaces form an API having a set of methods.

45. The method of claim 44 where the interface of at least one of the service providers includes less than all of the methods in the set.

46. The method of claim 44 where the API includes at least a subset of OLE DB.

47. The method of claim 41 further comprising replacing any of the service providers with another client having the same versioning interface.

48. The method of claim 41 further comprising adding another service provider having the same versioning interface.

49. The method of claim 41 further comprising replacing the client with a third-party client having the same versioning interface.

50. A medium bearing representations of instructions and data for causing a suitably programmed computer to perform the method comprising:

- sending commands from a client through a versioning interface;
- receiving the commands in a plurality of service providers all having an interface compatible with the versioning interface;
- executing the commands in the service providers;
- in the course of executing the commands, retrieving and storing a plurality of development documents in a version store having an interface compatible with the versioning interface.

Daryl

51. A method for developing software in a programmed digital computer, comprising:
receiving a user command from a client having a versioning interface in one of a plurality of service providers each having an interface compatible with the versioning interface;
executing an operation in response to the user command;
communicating a result through the compatible interface;
repeating the above steps for others of the service providers.

52. The method of claim 51 where executing the operation includes retrieving a development document from a version store having an interface compatible with the interface of the service providers.

53. The method of claim 51 where executing the operation comprises processing queries.

54. The method of claim 53 where the queries are directed to a version store containing versioned development documents.

55. The method of claim 51 where executing the operation comprises managing private copies of development documents.

56. A medium bearing representations of instructions and data for causing a suitably programmed computer to perform the method comprising:
receiving a user command from a client having a versioning interface in one of a plurality of service providers each having an interface compatible with the versioning interface;
executing an operation in response to the user command;
communicating a result through the compatible interface;
repeating the above steps for others of the service providers.

DRAFT EDITION 2010

57. A method for developing software in a programmed digital computer, comprising:
receiving a request for merging a plurality of development documents;
selecting one of a plurality of merge preprocessors;
modifying at least one of the development documents in the selected
preprocessor;
merging the development documents in a merge engine to produce a merged
document.

58. The method of claim 57 further comprising:
selecting one of a plurality of merge postprocessors;
modifying the output document in the selected postprocessor.

59. The method of claim 57 where the one preprocessor is selected in response to a
parameter in the request for merging.

60. The method of claim 57 where the input documents include both content and
properties.

61. The method of claim 60 where the merge engine merges the content separately from
the properties.

62. A medium bearing representations of instructions and data for causing a suitably
programmed computer to perform the method comprising:
receiving a request for merging a plurality of development documents;
selecting one of a plurality of merge preprocessors;
modifying at least one of the development documents in the selected
preprocessor;
merging the development documents in a merge engine to produce a merged
document.

Marked

63. A method for developing software in a programmed digital computer, comprising:
receiving a request for keyword expansion in a development document;
receiving a parameter specifying one of a plurality of keyword domains;
selecting one of a plurality of keyword expanders in response to the parameter;
expanding keywords in the document in the selected expander.

64. The method of claim 63 further comprising:
determining that keyword expansion is required;
selecting a generic keyword expander rather than any of the plurality of keyword expanders;
expanding keywords in the document in the generic expander.

65. The method of claim 64 where the generic expander expands the keywords in accordance with the keyword-domain parameter.

66. A medium bearing representations of instructions and data for causing a suitably programmed computer to perform the method comprising:
receiving a request for keyword expansion in a development document;
receiving a parameter specifying one of a plurality of keyword domains;
selecting one of a plurality of keyword-expansion service providers in response to the parameter;
expanding keywords in the document in the selected service provider.